PERL CORNER

Hacking CoinStar

by ntheory

CoinStar is a network of "self-service coin counting machines located at the front entrances of leading supermarkets nationwide" according to their corporate website. CoinStar machines will count your change and spit out a receipt telling you how much money you tossed in (minus their service charge which is currently 8.9% in the US). Take the receipt to the register and they'll give you cash so you don't have to count and roll your change.

When CoinStar started several years ago a few friends and I thought it was a great way to annoy the cashiers we didn't like. We'd go to the machine, throw a penny in, go up to the register with the receipt, and get the penny back from a very irritated cashier. It wasn't until this year that I realized I could peer into how it worked and see what was behind it and how it might be vulnerable.

I went out to the supermarket and started the very slow, methodical process of reverse engineering the receipts. I threw in a penny and got the receipt, then I did it again, then with two pennies, three pennies... by the end of it I had about 10 - 15 receipts that were just loaded with data I could use.

When I got home I looked at the receipts and noticed several things. First I noticed (obviously) the barcode. I scanned the receipt for numbers that matched the numbers in the barcode and I found the following:

- The first three digits were always "040"
- The next four digits are the transaction ID (this is also located on the bottom of the receipt)
- The next five digits represent the amount of money you put in in pennies
- The last digit looked like it could be a checksum digit of some sort

I couldn't figure out the pattern in the last digits so I hopped online and started looking up barcode formats. After a few minutes I came across a webpage that had a very in depth explanation of EAN-13 (http://www.barcodeisland.com/ean13.phtml). EAN-13 is the barcode format used in the US and Europe in most retail stores. The format is very well defined and almost every barcode scanner you'll see today can read them. A normal EAN-13 barcode is broken up into four sections:

- Number system (first two digits)
- Manufacturer code (next five digits)
- Product code (next five digits)
- Check digit (last digit)

Now I had enough information to generate that last digit so I wrote a script to generate the check digit for a receipt and tried it on my CoinStar data. To my surprise the first receipt didn't match and neither did the second. I realized that CoinStar borrowed the EAN-13 specification but broke the check digit scheme by implementing their own. ::sigh:: Something else to reverse engineer...

Let's take a look at one of my one penny barcodes:

0 409705 000017
A BCDEFG HIJKLM